

EE4800/EE5880

Real Time Embedded Systems

Lesson 21-24

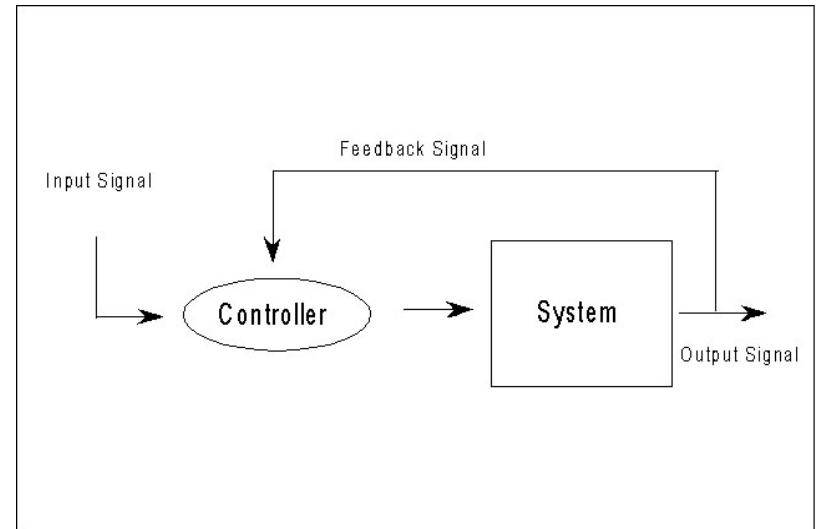
Fuzzy Logic

Overview

- What is a control system?
- Traditional vs Fuzzy Logic Control Systems
- Fuzzy Logic Controller Overview
- 68HC12 Fuzzy Logic Operations
- Membership Function Design Process
- Fuzzy Rule Implementation
- 68HC12 Fuzzification Process
- Rule Evaluation
- 68HC12 Defuzzification Process
- Mobile Robot Example

What is a control system?

- Any system, whose outputs are controlled by some inputs to the system, is called a control system
- Typical control system, Figure 4.1, pg 120



What is a control system?

- Uses model along with feedback provided by system output to generate a new system input
- Controller requires accurate mathematical model of system
 - Requires dynamic equations
 - Euler Lagrange Equation - Ref. Section 4.2.2.
 - Not fun stuff!!!

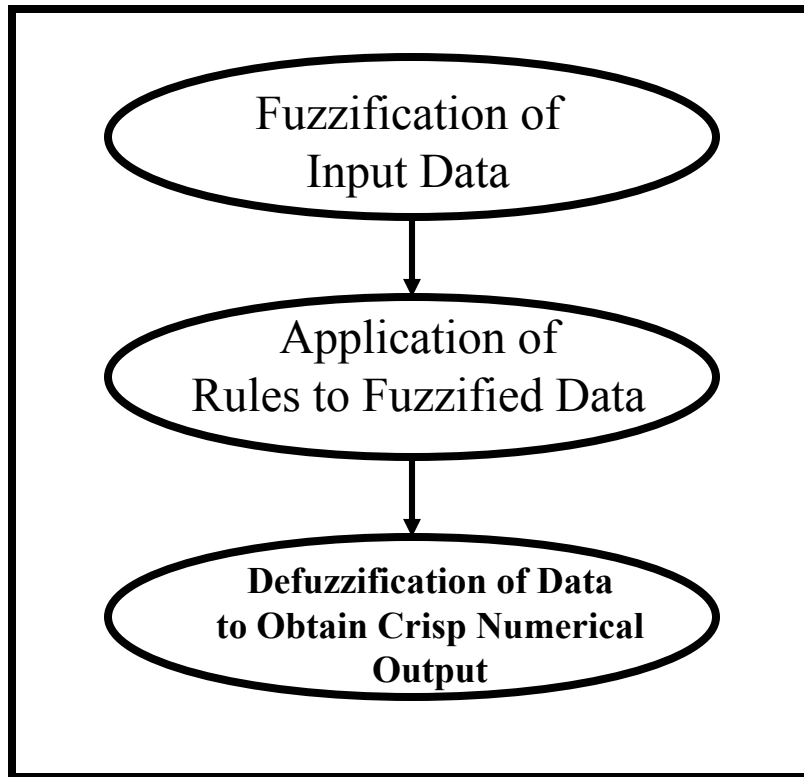
Traditional vs Fuzzy Logic Control Systems

- Traditional control method requires accurate, often complex system of equations
- Fuzzy logic proponents claim fuzzy systems are much closer to human control process
- Neural networks, fuzzy logic, and evolutionary systems share common thread
 - Do not rely on mathematical systems descriptions
- Interestingly, biological control systems do not rely on complex mathematical models
- Traditional camp treats these new approaches with skepticism for lack of mathematical rigor
- Fuzzy logic controllers in heavy use

Fuzzy Logic Controller Overview

- Fuzzy Logic Control Systems rely on physical rules rather than mathematical equations
- Simple form of an expert system
- Expert system contains rules that map input commands to output system responses
- Also an evidence accumulating system
 - Uses confidence factor to assign a system output according to the confidence measure computed using a well-defined process

Fuzzy Logic Control



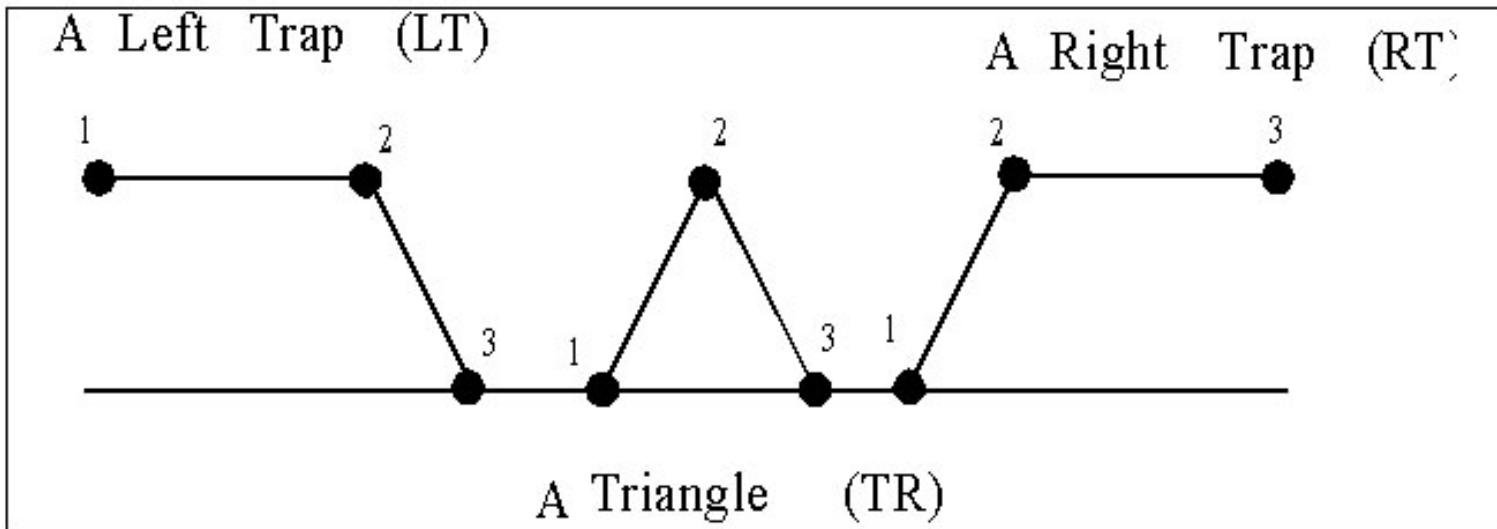
- Inputs should be sufficient to describe current status of system
- Need appropriate relationships between inputs and outputs
- Represent output control values in suitable form for system control

Fuzzy Logic Control

- Input values are first fuzzified by assigning an appropriate value (degree of membership) to each fuzzy membership function
- Fuzzy membership function is defined as an instance of a linguistic variable describing the fuzzy controller input
- All fuzzy membership functions have one of the following shapes: a left trap (LT), a trapezoid (TR) and a right trap (RT)

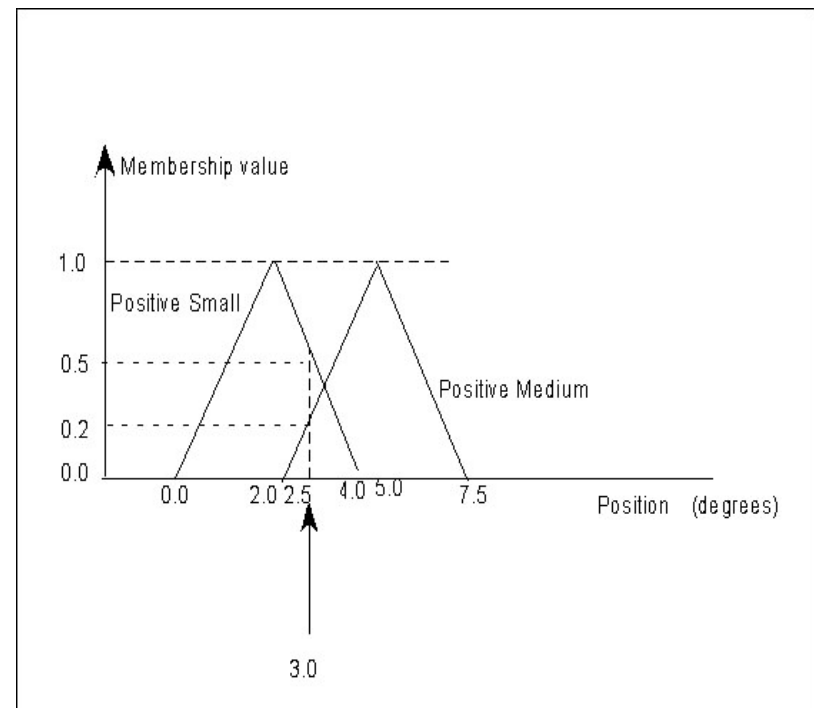
Fuzzy Logic Control

- HC12 implements all of its input membership functions using trapezoids
- Fig 4.5, page 128



Fuzzy Logic Control

- Given an input datum, the controller must first assign a degree of membership, a value ranging from zero to one, to each fuzzy membership function.
- Fig. 4.6, pg 129



Fuzzy Logic Control

- Apply expert rules to input membership functions
- Rules should cover all possible combinations of the input variables
- Rules map combinations of inputs to output force fuzzy membership functions
- Rule format:

If (antecedent 1) AND (antecedent 2),
Then (desired response)

Fuzzy Logic Control

- In applying rules, the **minimum** degree of membership among the input functions is assigned to the output membership function
 - Known as minimum rule
- Forms confidence factor
- Minimum of the two is chosen since it guarantees that both antecedents are true

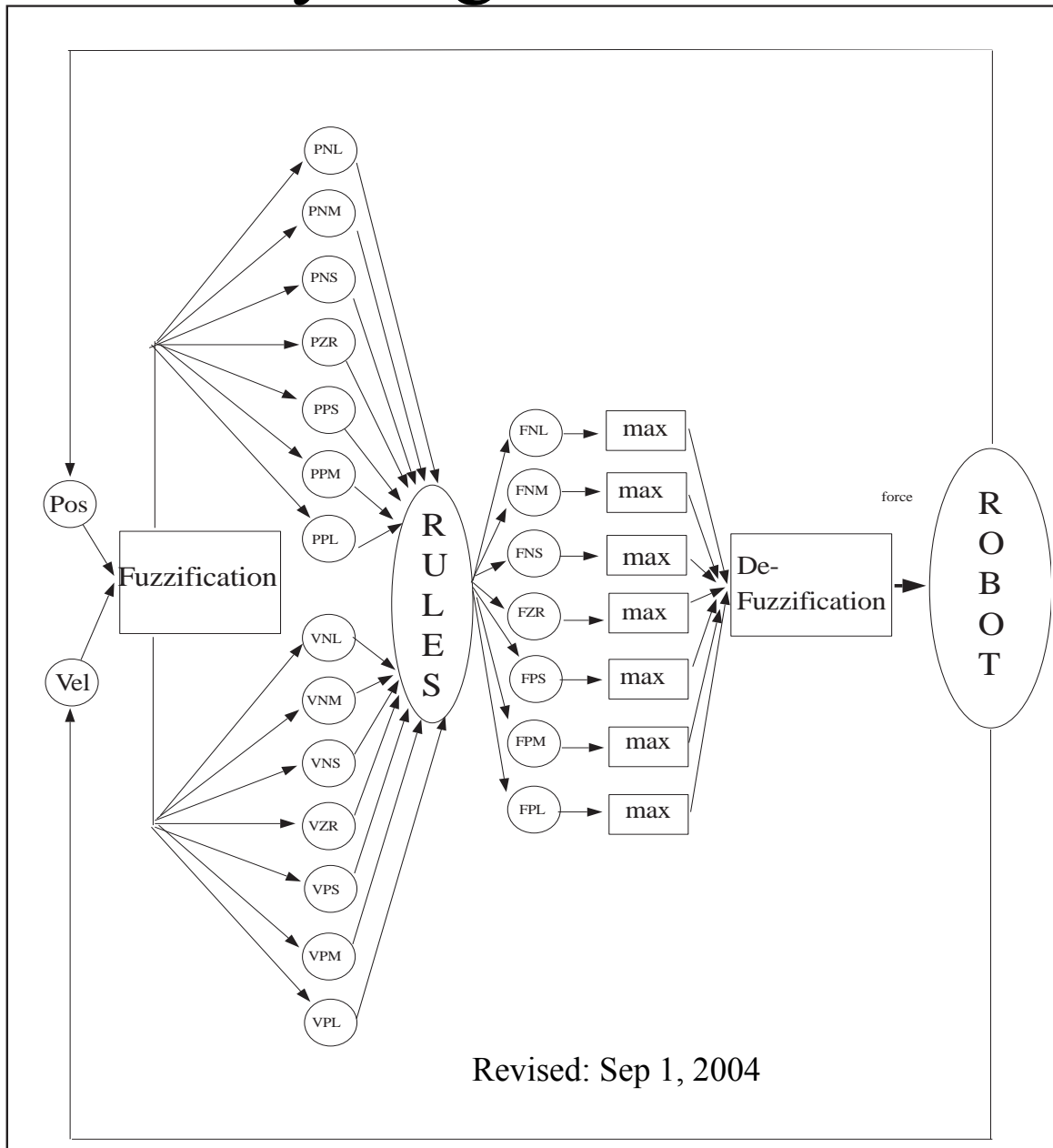
Fuzzy Logic Control

- Controller now takes the maximum of the confidence factors as the degree of the membership for the particular fuzzy function.
- Relies on the most dominant rule to assign the output membership function
- Need to convert the output membership function values to a form suitable to control system
 - Control value must be a numerical one
 - Need to reverse process of fuzzification - defuzzification

Fuzzy Logic Control

- Defuzzification – most common method is finding center of gravity or centroid of output membership functions
 - Combines all regions of the output fuzzy functions with OR operator
 - Computes center point
 - Generates crisp numerical value
- Gives equal weights to each output membership function

Fuzzy Logic Control



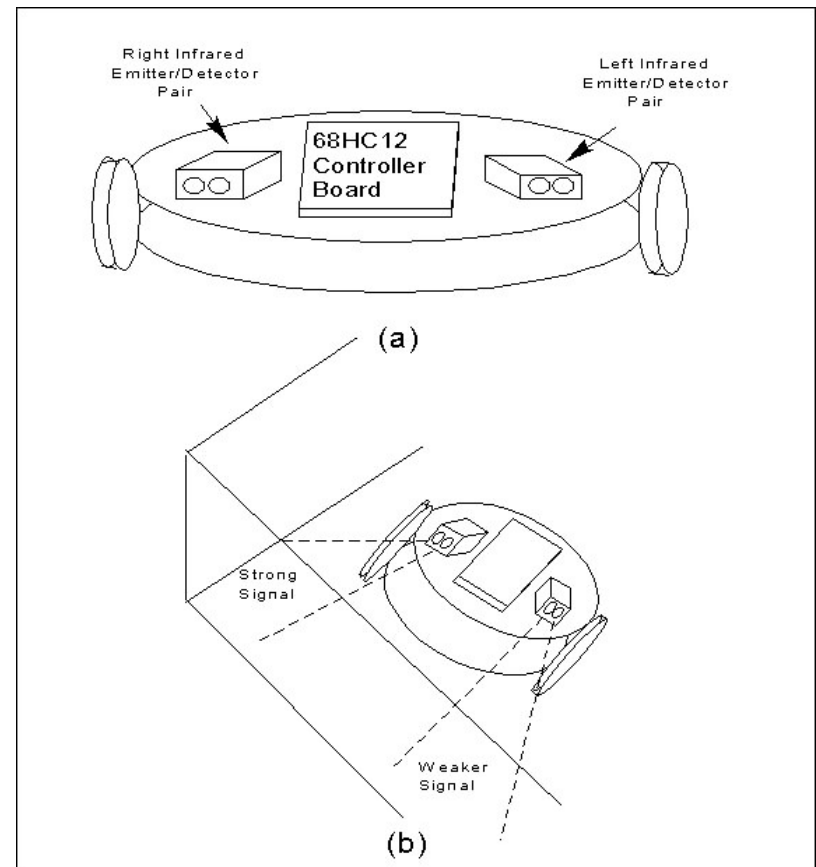
Revised: Sep 1, 2004

68HC12 Fuzzy Logic Operations

- Four 68HC12 instructions:
 - MEM: grade of membership
 - REV: MIN-MAX rule evaluation
 - REVW: MIN-MAX rule evaluation with optional rule weighting
 - WAV: performs weighted average calculations
- Supplementary instructions:
 - ETBL, TBL, EDIV, EDIVS, EMACS, EMAX, EMAXM, EMIND, EMINM, EMUL, EMULS

68HC12 Fuzzy Logic Operations

- Example: Design a fuzzy logic based robot controller - Fig 4.15, pg 149



68HC12 Fuzzy Logic Operations

- 1 . Determine linguistic variables describing all inputs to the controller

left sensor input: very weak, weak, medium, strong, very strong

right sensor input: very weak, weak, medium, strong, very strong

- 2 . Each linguistic variable must be defined with a set of input membership functions. For the HC12 a trapezoidal function must be used. Fig 4.16, pg 150

68HC12 Fuzzy Logic Operations

3 . Input membership function design process.

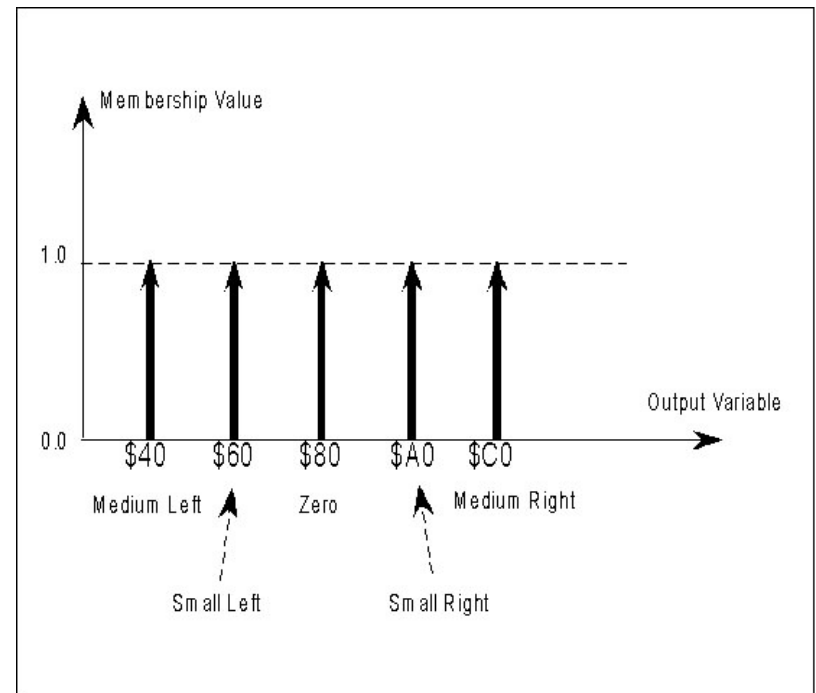
- Each membership function must be trapezoid: LT, T, RT
- Specify uniquely with farthest left point, farthest right point, left slope, and right slope
- Full membership corresponds to \$FF - for each sensor

<i>very weak</i>	<i>\$00, \$50, \$00, \$10</i>
<i>weak</i>	<i>\$40, \$70, \$10, \$10</i>
<i>medium</i>	<i>\$60, \$A0, \$10, \$10</i>
<i>strong</i>	<i>\$90, \$C0, \$10, \$10</i>
<i>very strong</i>	<i>\$B0, \$FF, \$10, \$00</i>

68HC12 Fuzzy Logic Operations

- 4 . Output membership function design process.
- Use singleton functions for the fuzzy output membership functions (Singleton: each function is defined by a single value).
 - *Medium Left* (\$40)
 - *Small Left* (\$60)
 - *Zero* (\$80)
 - *Small Right* (\$A0)
 - *Medium Right* (\$C0)
 - **Assumption:** \$80 keeps robot on straight path.

Figure 4.17, pg 151



68HC12 Fuzzy Logic Operations

5. Determine fuzzy rules that map input to output membership functions. Rules are intuitive based on considering all possible input sensor value scenarios.

Rule format:

IF (*antecedent 1*) AND (*antecedent 2*),
THEN (*desired output*)

68HC12 Fuzzy Logic Operations

Fuzzy Rules for Navigating Robot

Left/Right	Very Strong	Strong	Medium	Weak	Very Weak
Very Strong	<i>Zero</i>	<i>Small Right</i>	<i>Small Right</i>	<i>Medium Right</i>	<i>Medium Right</i>
Strong	<i>Small Left</i>	<i>Zero</i>	<i>Small Right</i>	<i>Medium Right</i>	<i>Medium Right</i>
Medium	<i>Small Left</i>	<i>Small Left</i>	<i>Zero</i>	<i>Small Right</i>	<i>Small Right</i>
Weak	<i>Medium Left</i>	<i>Medium Left</i>	<i>Small Left</i>	<i>Zero</i>	<i>Small Right</i>
Very Weak	<i>Medium Left</i>	<i>Medium Left</i>	<i>Small Left</i>	<i>Small Left</i>	<i>Zero</i>

68HC12 Fuzzy Logic Operations

6. Fuzzy Rule Implementation - encode rules for all input combinations using the format:

RULE 1:

antecedent one, antecedent two, \$FE, desired output, \$FE

Note: \$FE separator indicates convenient break points from fuzzy logic processing should an interrupt occur.

68HC12 Fuzzy Logic Operations

7. Defuzzify: Combine truth values of all output fuzzy membership functions into a single numerical value.

- Average the output function values - find the centroid of arbitrary output membership functions for singleton output membership functions.

$$\text{Output} = \frac{\sum (S_i F_i)}{\sum (F_i)} \quad \text{for } i = 1 \dots n$$

where: S_i is the Singleton value

F_i is the output membership value for fuzzy logic output function i

Mobile Robot Example Code

```
Line 1  ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
Line 2  ;Fuzzy Logic Controller for A Mobile Robot
Line 3  ;Description: This program takes in two Infrared
Line 4  ;sensor values and computes a di-
Line 5  ;rection control signal to avoid
Line 6  ;wall collisions.  The two sensor
Line 7  ;values are read from memory
Line 8  ;locations $6000 and $6001 and
Line 9  ;the control output value is
Line 10 ;written to memory location $6002
Line 11 ;
Line 12 ; Authors: Daniel Pack and Steve Barrett
Line 13 ; Date: 8-21-2000
Line 14 ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
```

Mobile Robot Example Code

```
Line 15 ; Data Section
Line 16 O_R_VS      EQU    $00    ;Offset values input and output mem fns
Line 17 O_R_ST      EQU    $01    ; Right Sensor Strong
Line 18 O_R_ME      EQU    $02    ; Right Sensor Medium
Line 19 O_R_WE      EQU    $03    ; Right Sensor Weak
Line 20 O_R_VW      EQU    $04    ; Right Sensor Very Weak
Line 21 O_L_VS      EQU    $05    ; Left Sensor Very Strong
Line 22 O_L_ST      EQU    $06    ; Left Sensor Strong
Line 23 O_L_ME      EQU    $07    ; Left Sensor Medium
Line 24 O_L_WE      EQU    $08    ; Left Sensor Weak
Line 25 O_L_VW      EQU    $09    ; Left Sensor Very Weak
Line 26 O_ML        EQU    $0A    ; Medium Left
Line 27 O_SL        EQU    $0B    ; Small Left
Line 28 O_ZR        EQU    $0C    ; Zero
Line 29 O_SR        EQU    $0D    ; Small Right
Line 30 O_MR        EQU    $0E    ; Medium Right
Line 31 MARKER      EQU    $FE    ; rule separator
Line 32 ENDR        EQU    $FF    ; end of rule marker
```

Mobile Robot Example Code

```
Line 33          ORG    $6000
Line 34 RSENSOR  RMB    $01    ; Allocating
Line 35 LSENSOR  RMB    $01    ; memory locations
Line 36 CONTROLS RMB    $01    ; for input/output variables

Line 37 ; Fuzzy Input Membership Function Definitions for Right Sensor
Line 38 R_Very_Strong  FCB    $B0, $FF, $10, $00
Line 39 R_Strong      FCB    $90, $C0, $10, $10
Line 40 R_Medium      FCB    $60, $A0, $10, $10
Line 41 R_Weak        FCB    $40, $70, $10, $10
Line 42 R_Very_Weak   FCB    $00, $50, $00, $10

Line 43 ; Fuzzy Input Membership Function Definitions for Left Sensor
Line 44 L_Very_Strong  FCB    $B0, $FF, $10, $00
Line 45 L_Strong      FCB    $90, $C0, $10, $10
Line 46 L_Medium      FCB    $60, $A0, $10, $10
Line 47 L_Weak        FCB    $40, $70, $10, $10
Line 48 L_Very_Weak   FCB    $00, $50, $00, $10
```

Mobile Robot Example Code

Line 49 ;Fuzzy Output Membership Function Definitions

Line 50 Medium_Left FCB \$40

Line 51 Small_Left FCB \$60

Line 52 Zero FCB \$80

Line 53 Small_Right FCB \$A0

Line 54 Medium_Right FCB \$C0

Line 55 ; Locations for fuzzy membership values

Line 56 R_VS RMB \$01

Line 57 R_ST RMB \$01

Line 58 R_ME RMB \$01

Line 59 R_WE RMB \$01

Line 60 R_VW RMB \$01

Line 61 L_VS RMB \$01

Line 62 L_ST RMB \$01

Line 63 L_ME RMB \$01

Line 64 L_WE RMB \$01

Line 65 L_VW RMB \$01

Mobile Robot Example Code

```
Line 66 ; Output Fuzzy Logic Membership Values - initialize them to zero
Line 67 ML          FCB    $00
Line 68 SL          FCB    $00
Line 69 ZR          FCB    $00
Line 70 SR          FCB    $00
Line 71 MR          FCB    $00
```

Mobile Robot Example Code

```
Line 72 ; Rule Definitions
Line 73 Rule_Start          FCB      O_R_VS,O_L_VS,MARKER,O_ZR,MARKER
Line 74                    FCB      O_R_VS,O_L_ST,MARKER,O_SL,MARKER
Line 75                    FCB      O_R_VS,O_L_MD,MARKER,O_SL,MARKER
Line 76                    FCB      O_R_VS,O_L_WE,MARKER,O_ML,MARKER
Line 77                    FCB      O_R_VS,O_L_VW,MARKER,O_ML,MARKER
Line 78                    FCB      O_R_ST,O_L_VS,MARKER,O_SR,MARKER
Line 79                    FCB      O_R_ST,O_L_ST,MARKER,O_ZR,MARKER
Line 80                    FCB      O_R_ST,O_L_MD,MARKER,O_SL,MARKER
Line 81                    FCB      O_R_ST,O_L_WE,MARKER,O_ML,MARKER
Line 82                    FCB      O_R_ST,O_L_VW,MARKER,O_ML,MARKER
Line 83                    FCB      O_R_MD,O_L_VS,MARKER,O_SR,MARKER
Line 84                    FCB      O_R_MD,O_L_ST,MARKER,O_SR,MARKER
Line 85                    FCB      O_R_MD,O_L_MD,MARKER,O_ZR,MARKER
Line 86                    FCB      O_R_MD,O_L_WE,MARKER,O_SL,MARKER
Line 87                    FCB      O_R_MD,O_L_VW,MARKER,O_SL,MARKER
Line 88                    FCB      O_R_WE,O_L_VS,MARKER,O_MR,MARKER
Line 89                    FCB      O_R_WE,O_L_ST,MARKER,O_MR,MARKER
Line 90                    FCB      O_R_WE,O_L_MD,MARKER,O_SR,MARKER
Line 91                    FCB      O_R_WE,O_L_WE,MARKER,O_ZR,MARKER
Line 92                    FCB      O_R_WE,O_L_VW,MARKER,O_SL,MARKER
Line 93                    FCB      O_R_VW,O_L_VS,MARKER,O_MR,MARKER
Line 94                    FCB      O_R_VW,O_L_ST,MARKER,O_MR,MARKER
Line 95                    FCB      O_R_VW,O_L_MD,MARKER,O_SR,MARKER
Line 96                    FCB      O_R_VW,O_L_WE,MARKER,O_SR,MARKER
Line 97                    FCB      O_R_VW,O_L_WE,MARKER,O_ZR,ENDR
```

Mobile Robot Example Code

```
Line 98 ; Main Program
Line 99          ORG          $4000

Line 100 ; Fuzzification-----
; Fuzzification Procedure
; MEM function determines grade of membership.
;
; Registers must be set up as follows:
; A: must hold the current crisp value of system input variable
;   X: must point to 4-byte data structure describing the trapezoidal
;     membership function for a label of the system input
;   Y: must point to the fuzzy input (RAM location) where the resulting
;     grade of membership is to be stored
;-----

; Perform fuzzification for right sensor

Line 101          LDX      #R_Very_Strong      ; Start of Input Mem func
Line 102          LDY      #R_VS              ; Start of Fuzzy Mem values
Line 103          LDAA     RSENSOR             ; Right Sensor Value
Line 104          LDAB     #5                 ; Number of iterations
Line 105          Loopr   MEM                  ; Assign mem value
Line 106          DBNE    B,Loopr             ; Do all five iterations
```

Mobile Robot Example Code

```
        ; Perform fuzzification for left sensor
Line 107      LDAA  LSENSOR          ; Left Sensor Value
Line 108      LDAB  #5              ; Number of iterations
Line 109      Loop1 MEM            ; Assign mem value
Line 110      DBNE  B,Loop1        ; Do all five iterations
```


Mobile Robot Example Code

```
; Fuzzy Logic Rule Evaluation-----  
; REV function performs an unweighted evaluation of a list of rules,  
; using fuzzy input values to produce fuzzy outputs.  
;  
; Before executing REV, perform the following set up operations.  
; X: must point to the first 8-bit element in the rule list.  
;   Y: must point to the base address for fuzzy inputs and outputs  
;   A: must contain the value of $FF, and the CCR V bit must = 0  
;     (LDAA #$FF places the correct value in A and clears V)  
;   Clear fuzzy outputs to zeros  
;-----
```

```
Line 111      LDY    #R_VS          ; Process rules  
Line 112      LDX    #Rule_Start    ; Point X to the start addr  
Line 113      LDAA   #$FF          ; Initialize min and V bit  
Line 114      REV                      ; Evaluate rules
```

Mobile Robot Example Code

```
Line 115 ; Defuzzification Process-----  
; WAV function performs weighted average calculations on  
; values stored in memory.  
; X: used to reference one source operand list  
; Y: used to reference second source operand list  
; B: used a s counter to control number of elements to  
; be included in the weighted average  
;-----  
Line 116 LDX #Medium_Left ; Start of output mem func  
Line 117 LDY #ML ; Start of mem values  
Line 118 LDAB #$05 ; Five elements sum  
Line 119 WAV ; Computing a crisp value  
Line 120 EDIV ; Extended divide - 32 bit/16 bit  
Line 121 TFR Y,D ; Store answer to D  
Line 122 STAB CONTROLS ; Save the answer  
Line 123 END
```